

## Test 7

```
class BankAccount {  
    private String accountNumber;  
    private double balance;  
  
    public BankAccount(String accountNumber, double balance) {  
        this.accountNumber = accountNumber;  
        this.balance = balance;  
    }  
  
    public void deposit(double amount) {  
        if (amount > 0) {  
            balance += amount;  
            System.out.println("Deposited " + amount + ". New balance: " + balance);  
        } else {  
            System.out.println("Deposit amount must be positive.");  
        }  
    }  
  
    public void withdraw(double amount) {  
        if (amount > 0 && amount <= balance) {  
            balance -= amount;  
            System.out.println("Withdrew " + amount + ". New balance: " + balance);  
        } else {  
            System.out.println("Withdrawal denied: Insufficient funds or invalid amount.");  
        }  
    }  
  
    public double getBalance() {
```

```

        return balance;
    }

    public String getAccountNumber() {
        return accountNumber;
    }
}

class SavingsAccount extends BankAccount {
    private static final double MIN_BALANCE = 1000.0;
    private static final double INTEREST_RATE = 0.03; // 3% interest

    public SavingsAccount(String accountNumber, double balance) {
        super(accountNumber, balance);
    }

    @Override
    public void withdraw(double amount) {
        if (getBalance() - amount < MIN_BALANCE) {
            System.out.println("Withdrawal denied: Minimum balance requirement not met.");
        } else {
            super.withdraw(amount);
        }
    }

    public void applyInterest() {
        double interest = getBalance() * INTEREST_RATE;
        deposit(interest);
        System.out.println("Applied interest. New balance: " + getBalance());
    }
}

```

```

}

class CheckingAccount extends BankAccount {

    private static final double OVERDRAFT_LIMIT = 500.0;

    public CheckingAccount(String accountNumber, double balance) {
        super(accountNumber, balance);
    }

    @Override
    public void withdraw(double amount) {
        if (getBalance() - amount < -OVERDRAFT_LIMIT) {
            System.out.println("Withdrawal denied: Overdraft limit exceeded.");
        } else {
            super.withdraw(amount);
        }
    }
}

```

```

public class BankAccountDemo {

    public static void main(String[] args) {

        SavingsAccount savings = new SavingsAccount("SA12345", 1500.0);
        CheckingAccount checking = new CheckingAccount("CA12345", 500.0);

        savings.deposit(200.0);
        savings.withdraw(600.0);
        savings.applyInterest();
        System.out.println("Savings Account Balance: " + savings.getBalance());

        checking.deposit(100.0);
    }
}

```

```

        checking.withdraw(600.0);

        System.out.println("Checking Account Balance: " + checking.getBalance());
    }
}

```

	Output
^	<pre> java -cp /tmp/CjIruvm2fV/BankAccountDemo Deposited 200.0. New balance: 1700.0 Withdrew 600.0. New balance: 1100.0 Deposited 33.0. New balance: 1133.0 Applied interest. New balance: 1133.0 Savings Account Balance: 1133.0 Deposited 100.0. New balance: 600.0 Withdrew 600.0. New balance: 0.0 Checking Account Balance: 0.0  === Code Execution Successful === </pre>

```

abstract class GameCharacter {
    protected String name;
    protected int health;
    protected int level;

    public GameCharacter(String name, int health, int level) {
        this.name = name;
        this.health = health;
        this.level = level;
    }
}

```

```
}

public abstract void attack();

public abstract void defend();

public String toString() {
    return String.format("Name: %s, Health: %d, Level: %d", name, health, level);
}
}
```

```
class Warrior extends GameCharacter {
    public Warrior(String name, int health, int level) {
        super(name, health, level);
    }

    public void attack() {
        System.out.println(name + " swings a sword!");
    }

    public void defend() {
        System.out.println(name + " blocks with a shield!");
    }
}
```

```
class Mage extends GameCharacter {
    public Mage(String name, int health, int level) {
        super(name, health, level);
    }
}
```

```

    public void attack() {
        System.out.println(name + " casts a fireball!");
    }

    public void defend() {
        System.out.println(name + " casts a shield spell!");
    }
}

class Archer extends GameCharacter {
    public Archer(String name, int health, int level) {
        super(name, health, level);
    }

    public void attack() {
        System.out.println(name + " shoots an arrow!");
    }

    public void defend() {
        System.out.println(name + " dodges the attack!");
    }
}

public class GameCharacterDemo {
    public static void main(String[] args) {
        Warrior warrior = new Warrior("sai", 100, 5);
        Mage mage = new Mage("luffy", 80, 7);
        Archer archer = new Archer("zoro", 90, 6);
    }
}

```

```
        System.out.println(warrior);  
        warrior.attack();  
        warrior.defend();  
  
        System.out.println(mage);  
        mage.attack();  
        mage.defend();  
  
        System.out.println(archer);  
        archer.attack();  
        archer.defend();  
    }  
}
```

## Output

```
^ java -cp /tmp/gOW5FRGMR9/GameCharacterDemo  
Name: sai, Health: 100, Level: 5  
sai swings a sword!  
sai blocks with a shield!  
Name: lufyy, Health: 80, Level: 7  
lufyy casts a fireball!  
lufyy casts a shield spell!  
Name: zoro, Health: 90, Level: 6  
zoro shoots an arrow!  
zoro dodges the attack!  
  
=== Code Execution Successful ===
```

```
abstract class Product {  
    private String productId;  
    private String name;  
    private double price;  
  
    public Product(String productId, String name, double price) {  
        this.productId = productId;  
        this.name = name;  
        this.price = price;  
    }  
  
    public abstract double calculateDiscount(boolean isMemberOrSale);  
  
    public String getProductId() {  
        return productId;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public double getPrice() {  
        return price;  
    }  
  
    public String toString() {  
        return String.format("ID: %s, Name: %s, Price: %.2f", productId, name, price);  
    }  
}
```



```

class Electronics extends Product {
    private static final double MEMBER_DISCOUNT = 0.10; // 10% discount for members

    public Electronics(String productId, String name, double price) {
        super(productId, name, price);
    }

    public double calculateDiscount(boolean isMember) {
        double discount = isMember ? MEMBER_DISCOUNT : 0;
        double discountedPrice = getPrice() * (1 - discount);
        System.out.println("Electronics Price (Member: " + isMember + "): " + discountedPrice);
        return discountedPrice;
    }
}

class Clothing extends Product {
    private static final double SEASONAL_SALE_DISCOUNT = 0.20;

    public Clothing(String productId, String name, double price) {
        super(productId, name, price);
    }

    public double calculateDiscount(boolean isSeasonalSale) {
        double discount = isSeasonalSale ? SEASONAL_SALE_DISCOUNT : 0;
        double discountedPrice = getPrice() * (1 - discount);
        System.out.println("Clothing Price (Seasonal Sale: " + isSeasonalSale + "): " + discountedPrice);
        return discountedPrice;
    }
}

```

```
}
```

```
public class ProductDemo {  
    public static void main(String[] args) {  
        Electronics laptop = new Electronics("54erevrfre", "Laptop", 1000.0);  
        Clothing shirt = new Clothing("C456", "Shirt", 50.0);  
  
        laptop.calculateDiscount(true);  
        laptop.calculateDiscount(false);  
  
        shirt.calculateDiscount(true);  
        shirt.calculateDiscount(false);  
    }  
}
```

## Output

```
java -cp /tmp/LJoLau9y0V/ProductDemo  
Electronics Price (Member: true): 900.0  
Electronics Price (Member: false): 1000.0  
Clothing Price (Seasonal Sale: true): 40.0  
Clothing Price (Seasonal Sale: false): 50.0  
  
=== Code Execution Successful ===
```

```
// Base class
```

```
abstract class LibraryItem {  
    private String title;  
    private String author;
```

```
private int year;
```

```
public LibraryItem(String title, String author, int year) {  
    this.title = title;  
    this.author = author;  
    this.year = year;  
}
```

```
public abstract void checkIn();  
public abstract void checkOut();
```

```
public String toString() {  
    return String.format("Title: %s, Author: %s, Year: %d", title, author, year);  
}  
}
```

```
class Book extends LibraryItem {  
    private boolean isCheckedOut;
```

```
public Book(String title, String author, int year) {  
    super(title, author, year);  
    this.isCheckedOut = false;  
}
```

```
public void checkIn() {  
    if (isCheckedOut) {  
        isCheckedOut = false;  
        System.out.println("Book \"" + super.toString() + "\" checked in.");  
    } else {
```

```

        System.out.println("Book \"" + super.toString() + "\" is checked in.");
    }
}

public void checkOut() {
    if (!isCheckedOut) {
        isCheckedOut = true;

        System.out.println("Book \"" + super.toString() + "\" checked out.\n");
    } else {
        System.out.println("Book \"" + super.toString() + "\" is checked out.\n");
    }
}
}

```

```

class DVD extends LibraryItem {
    private boolean isCheckedOut;

    public DVD(String title, String author, int year) {
        super(title, author, year);
        this.isCheckedOut = false;
    }

    public void checkIn() {
        if (isCheckedOut) {
            isCheckedOut = false;

            System.out.println("DVD \"" + super.toString() + "\" checked in.");
        } else {
            System.out.println("DVD \"" + super.toString() + "\" is checked in.");
        }
    }
}

```

```

    }

    public void checkOut() {
        if (!isCheckedOut) {
            isCheckedOut = true;

            System.out.println("DVD \"" + super.toString() + "\" checked out.");
        } else {
            System.out.println("DVD \"" + super.toString() + "\" is checked out.\n");
        }
    }
}

```

```

public class LibraryDemo {
    public static void main(String[] args) {
        Book book = new Book("Java Programming", "donlee", 2020);
        DVD dvd = new DVD("Inception", "trifagar law", 2010);

        System.out.println(book);

        book.checkIn();

        book.checkOut();

        System.out.println(dvd);
        dvd.checkIn();
        dvd.checkOut();
    }
}

```

## Output

Clear

```
^ java -cp /tmp/sdt9EGIm4/LibraryDemo
```

```
Title: Java Programming, Author: donlee, Year: 2020
```

```
Book "Title: Java Programming, Author: donlee, Year: 2020" is checked in.
```

```
Book "Title: Java Programming, Author: donlee, Year: 2020" checked out.
```

```
Title: Inception, Author: trifagar law, Year: 2010
```

```
DVD "Title: Inception, Author: trifagar law, Year: 2010" is checked in.
```

```
DVD "Title: Inception, Author: trifagar law, Year: 2010" checked out.
```

```
=== Code Execution Successful ===
```